Electronic Version
Stylesheet Version v1.1.1

# Description

# METHOD FOR TRANSFERRING A COMMAND AMONG A PLURALITY OF DEVICES IN A COMPUTER SYSTEM

## BACKGROUND OF INVENTION

[0001] 1. Field of the Invention

[0002] The invention relates to a method for transferring a command among a plurality of devices of a computer system, and more particularly, to a method for executing an examining process before the command is transferred to make sure the command is being correctly transferred and to quickly reboot the computer system.

[0003] 2. Description of the Prior Art

[0004] With the rapid development of information technology, computer systems play important roles in modernizing many companies and are widely used in practically every industry. Consequently, the manufacturing of portable computers, including PDAs and notebooks, has become a

mainstream business in the computer industry.In order to utilize the portable computer systems without an external power supply, users require batteries to provide sufficient power to the computer system. However, the power stored in the battery is exhaustible, so users usually turn off the portable computer systems when they do not use the portable computer systems, and the portable computer systems will be rebooted after the users arrive at a certain place and begin to work. However, since the graphical operating systems develop quickly, the more powerful operating systems become, the longer the portable computer systems take to load the related operating systems. That is, the reboot process becomes very complicated and time-consuming. It is very inconvenient for computer users to wait for a long time in the reboot process just in order to check a calendar or memorandum data.

[0005] For improving the above-mentioned lack of efficiency in the reboot process, we have to clarify how the computer system operates in the booting process. Currently, in the booting process, most computer systems utilize a basic input/output system (BIOS) program that is stored in a non-volatile memory, and which is first executed by a central processing unit (CPU) when the computer system

is turned on. Besides providing key low-level support for device drivers and operating systems, the BIOS program contains a power on self-test (POST) program, and a bootstrap program. The POST program ensures that the basic components of the computer system are operating correctly. After a successful completion of the POST program, the BIOS program executes the bootstrap program, the purpose of which is to load boot code into memory to boot an operating system.

[0006] In addition, regarding the notebook or PC system, a prior-art STR (suspend to RAM) feature has been applied for the computer entering into a standby mode from an operating mode. After being relieved from the standby mode(an STR status), the computer can quickly resume operation (back to the operating mode).It takes only 5 seconds from holding to recovering (from the standby mode to the operating mode). Please refer to Fig.1. Fig.1 is a functional block diagram of a (portable) computer system 10 according to the prior art. The computer system 10 includes a processor 12, a non-volatile storage device 14, a volatile storage device 16, and a periphery device module 18. The non-volatile storage device14 is used to store the BIOS of the computer system 10, and the processor 12 is used to

control operations of the computer system 10. In addition, the processor 12 can exchange instructions and signals with the non-volatile storage device14, the volatile storage device 16, a display processing module 34, and the periphery device module 18 so as to fully operate the computer system 10.

[0007] When the computer system 10 operates, the computer system 10 will load the related codes, including the operating system, the programming codes, and part of the BIOS into the volatile storage device 16, so that the processor 12 can execute those codes to operate the computer system 10. When users utilize the prior-art STR function, users can press at least a predetermined button of the computer system 10 to switch the computer system 10 from the operating mode to the standby mode. After users press the predetermined button, the processor 12 will confirm that the computer system 10 is about to enter into the standby mode according to a basic code of the BIOS. Since the volatile storage device 16 stores the codes including a condition describing how the computer system 10 operates before the standby mode, the computer system 10 stops providing power to the processor 12 and the periphery device module 18 and provides power only to

the volatile storage device 16 so that the volatile storage device 16 can maintain the codes. When the computer system 10 is in the standby mode, the processor 12 and the periphery device module 18 stop operating and consume little power. In the standby mode, due to that the operations of the processor 12, the periphery device module 18, and most devices and circuits of the computer system 10 are terminated, the power demands for the computer system 10 will be largely reduced, which is the objective of the STR function (the standby mode).

[0008] Corresponding to the above-mentioned operation directing the computer to enter into the standby mode, the computer system 10 must include at least a related function for waking up the computer system 10 (a plurality of codes or signals related to wake-up events can be stored in the BIOS in the non-volatile storage device14). Generally, the action of pushing any button of the computer system 10 or touching aturn-on key matches a wake-up event directing the computer system 10 from the standby mode to the operating mode. When the computer system 10 is about to be switched from the standby mode to the operating mode, the computer system 10 will execute the BIOS stored in the non-volatile storage device14 to leave

the standby mode. At this moment, the command controlling the computer system 10 is located in the non-volatile storage device14. After part of the BIOS stored in the non-volatile storage device14 is executed, the execution step will jump to the volatile storage device 16 so that the codes (the BIOS or the operating system) stored in the volatile storage device 16 will continue to achieve the mode-transformation process (from the standby mode to the operating mode). At this moment, the command (mainly controlling the computer system 10) is transferred from the non-volatile storage device14 to the volatile storage device 16. In the standby mode, the volatile storage device 16 can maintain the stored codes (due to that power still supplies the volatile storage device 16 in the standby mode), and the stored codes of the volatile storage device 16 include a condition describing how the computer system 10 operates before entering in the standby mode. Therefore, once the computer system 10 is relieved from the standby mode, the processor 12 can directly execute the BIOS or the operating system in the volatile storage device 16 and quickly continue the operations of the computer system 10.

[0009] Therefore, the command transferring process plays an

important role when the computer system operates, especially during the boot/reboot process. The result of the command transferring process determines whether the computer system can normally and smoothly operate or not. Please continue to refer to Fig.1. When the BIOS or the operating system in the volatile storage device 16 is incorrect, non-intact, or even non-existent, the operation may fail and the computer system 10 may even crash after the command is transferred from the non-volatile storage device14 to the volatile storage device 16. For illustrating importance of the command transferring process when the computer system operates, please refer to Fig.2, which is a functional block diagram of an embodiment showing operations of partial devices shown in Fig.1. The present embodiment shown in Fig.2 includes the processor 12 and the periphery device module 18. The periphery device module 18 includes a display device 20, and the display device 20 corresponds to an application code, which may be stored in a BIOS ofthe display device 20. When the processor 12 is utilized to operate the display device 20, the command should be transferred to the BIOS of the display device 20 to execute theapplication code. After theapplication code is partially or fully executed, the

display device 20 then returns the command back to the processor 12. Similar to the embodiment shown in Fig.1, if the BIOS or the application code (in the display device 20) is incorrect, non-intact, or even non-existent, the command transferring process may fail and the computer system 10 may even crash due to that there is no corresponding examining process to predictably ensure the accuracy of the command transferring process.

## SUMMARY OF INVENTION

[0010] It is therefore a primary objective of the claimed invention to provide a method for transferring a command among a plurality of devices of a computer systemto ensure the accuracy of a command transferring process and to solve the above-mentioned problems.

[0011] According to the claimed invention, amethod for transferring a command among a plurality of devices of a computer system to operate the computer system is disclosed. The plurality of devices at least include a first storage device and a second storage device, and the first storage device is used for storing a first code. The method includes (a)executing the first code in the first storage device; (b)after proceeding with step(a), executing an examining process before the first storage device transfers the

command to the second storage device; (c)after proceeding with step (b), the first storage device transferring thecommand to the second storage device to operate the computer system when a result of the examining process is correct; and (d)after proceeding with step (b), the first storage device not transferring the command to the second storage device when the result of the examining process is incorrect.

[0012] According to the claimed invention, a method for switching a computer system from a standby mode to an operating mode is disclosed. The computer system includes a first storage device for storing a basic input output system (BIOS) of the computer system, and a second storage device for temporarily storing data.The method includes: (a)utilizing the second storage device to store a condition before the computer system enters the standby mode; (b)executing the BIOS in the first storage device to cease the standby mode; (c)after proceeding with step (b), executing an examining process; and (d)after proceeding with step (c), switching the computer system into the operating mode according to the condition stored in the second storage device when the result of the examining process is correct.

[0013] According to the claimed invention, a method for transferring a command among plurality of devices in a computer system includes (a)executing an examining process when transferring the command; (b)in step (a), achieving the transfer of the command when a result of the examining process is correct; and (c)in step (a), failing the transfer of the command when the result of the examining process is incorrect.

[0014] These and other objectives of the present invention will no doubt become obvious to those of ordinary skill in the art after reading the following detailed description of the preferred embodiment, which is illustrated in the various figures and drawings.

## Brief Description of Drawings

[0015] Fig.1 is a functional block diagram of a computer system according to the prior art.

[0016] Fig.2 is a functional block diagram of an embodiment showing operations of partial devices shown in Fig.1.

[0017] Fig.3 is a functional block diagram of a computer system according to the present invention.

[0018] Fig.4 is a flow chart of an embodiment of the present invention.

[0019] Fig.5 is a functional block diagram of an embodiment that

inherits the structure of the computer system shown in Fig.3.

[0020] Fig.6 is a flow chart of another detailedmethod embodiment.

[0021] Fig.7 is a functional block diagram of another embodiment related to Fig.2.

[0022] Fig.8 is a flow chart showing the method for transferring a command among a plurality of devices of the computer system.

## DETAILED DESCRIPTION

[0023] One of the main characteristics of the present invention is an examining process executed before a computer system is switched from a standby mode to an operating mode so that the computer system can correctly achieve the mode transformation to be quicklyrebooted. The related hardware structure is shown in Fig.3, which is a functional block diagram of a computer system according to the present invention. The computer system 30 can be a notebook computer, a personal computer system (PC), an information appliance, or a PDA. The computer system 30 includes a first storage device 34 and a second storage device 36. The first storage device 34 can be a non-volatile read-only storage device (ROM), which is used to

store the BIOS of the computer system 30, and the second storage device 36 can be a volatile random access storage device (RAM), which is used to temporarily store a code. The stored code can be the BIOS, an operating system, or other programming code. The computer system 30 further includes a processor 32 used to control operations of the computer system 30. When the computer system 30 is in the operating mode, the computer system 30 will load necessary codes, including the operating system, programming codes, or even part of the BIOS, into the second storage device 36, so that the processor 32 can execute these codes to operate the computer system 30.

[0024] When the computer system 30 is about to be switched to the standby mode, the second storage device 36 stores codes including a condition describing how the computer system 30 operates before entering into the standby mode. After the computer system 10 enters into the standby mode, the computer system 10 stops providing power for the processor 32 and other circuitries except the second storage device 36 so that the second storage device 36 can maintain the stored codes. Therefore, due to that the processor 32 and most of the circuitries in the computer system 30 are terminated in the standby mode,

the computer system 30 consumes much less power in the standby mode than does the computer system 30 in the operating mode. Afterwards, if users want to switch the computer system 30 from the standby mode back to the operating mode, the BIOS in the first storage device 34 will be executed to terminate the standby mode and an examining process will be immediately executed. The examining process will be detailed in following paragraphs. Please notice that only when the result of the examining process is correct, the computer system 30 can be switched to the operating mode according to the codes (the condition of the computer system 30) stored in the second storage device 36. On the other hand, when the result of the examining process is incorrect, a re-boot process or a debug process will be executed to avoid executing the incorrect or non-intact codes in the second storage device 36. Anembodiment showing a method according to the present invention can refer to Fig.4, which is a flow chart of an embodiment of the present invention with following steps:

[0025] Step100:start;

[0026] Step102:utilize the second storage device 36 to store a condition describing how the computer system 30 oper-

ates before entering into the standby mode;

[0027] Step104:execute the BIOS in the first storage device 34 so as to terminate the standby mode;

[0028] Step106:execute the examining process; if the result of the examining process is correct, proceed with step 108; if the result of the examining process is incorrect, proceed with step 110;

[0029] Step108:execute the stored codes (the BIOS or the operating system) in the second storage device 36 according to the condition of the computer system 30 stored in the second storage device 36, and then switch the computer system 30 to the operating mode to operate the computer system 30;

[0030] Step110:do not switch the computer system 30 to the operating mode, and execute the re-boot process or debug process.

[0031] As the method described in Step102, when the computer system 30 executes the BIOS in the first storage device 34 to terminate the standby mode, the command controlling the computer system 30 is located in the first storage device 34. When a specific instruction of the BIOS in the first storage device 34 is executed, an execution step will jump to the second storage device 36 so that the stored codes

(BIOS or the operating system) in the second storage device 36 can be utilized to continue finishing the mode transformation. When the execution step will jump to the second storage device 36, the command is transferred from the first storage device 34 to the second storage device 36. Please refer to Fig.5, which is a functional block diagram of an embodiment that inherits the structure of the computer system 30 shown in Fig.3. Please notice that another register 42 is installed in the present embodiment shown in Fig.5. Due to that the content of the codes (BIOS or the operating system) in the second storage device 36 are previously available for users, users can determine whether the result of the examining process is correct or incorrect by examining whether a predetermined instruction of the codes conforms to a predetermined condition or not according to the content of the codes before the command is transferred from the first storage device 34 to the second storage device 36. For instance, a first instruction stored in the second storage device 36, such as "S3_First_Com", can be loaded into the register 42 shown in Fig.5 to compare if the loaded first instruction (S3_First_Com) matches a predetermined instruction. The loading and comparing processes can be implemented by

software or hardware of the computer system 30. Therefore, if the loaded first instruction matches the predetermined instruction (both are "S3_First_Com"), the result of the examining process is correct. On the other hand, if the result of the examining process is incorrect, and the computer system 30 has to be debugged or rebooted.

[0032] Based on the embodiment shown in Fig.5, Fig.6 shows a detailedmethod embodiment by comparing the first instruction stored in the second storage device 36 with a predetermined instruction to operate the computer system 30. Please refer to Fig.6, which is a flow chart of another detailedmethod embodiment, which includes following step:

[0033] Step200:start;

[0034] Step202:execute the BIOS in the first storage device 34;

[0035] Step204:load the first instruction of the codes stored in the second storage device 36 into a register 42 before the command is transferred from the first storage device 34 to the second storage device 36;

[0036] Step206:check whether the loaded predetermined instruction matches the predetermined instruction or not; if the loaded first instruction matches the predetermined in-

struction, proceed with Step 208; if the loaded first instruction does not match the predetermined instruction, proceed with Step 210;

[0037] Step208:transfer the command from the first storage device 34 to the second storage device 36, and execute the stored code (BIOS or the operating system) in the second storage device 36 so as to switch the computer system 30 to the operating mode;

[0038] Step210:do not transfer the command from the first storage device 34 to the second storage device 36, and do not switch the computer system 30 to the operating mode; afterwards, execute a re-boot process or a debug process.

[0039] The characteristics of the present invention are not constrained in the mode transformation of the computer system 30. In fact, the method of the present invention can be applied to various conditions whenever a command is transferred among a plurality of devices in the computer system 30. The characteristic of the present invention is to execute an examining process before the command is transferred so as to ensure that the command is transferred correctly. Please refer to Fig.7, which is a functional block diagram of another embodiment related to Fig.2. The present embodiment inherits the structure shown in

Fig.2 with an additional installation of a register 52. The present embodiment includes a processor 42 and a periphery device module 48. The periphery device module 48 is a display device 50. The display device 50 still includes corresponding BIOS of the display device and an application code with given content. Before the processor 42 actuates the display device 50 and transfers the command to the BIOS of the display device 50, an examining process can be executed according to a predetermined instruction or a predetermined tag in the present invention. According to the former statements in the previous embodiments, the predetermined tag, which can be a tag initially contained in the application code or later added by users, can be utilized for examinationin a manner similar to that of the predetermined instruction; that is, users or program designers can utilize the application code to load the predetermined tag in the given content of the application code in the display device 50 (such as INT10_SYMBOL) into the register 52 shown in Fig.7 before the command is transferred to the display device 50, and determines whether the loaded predetermined tag (INT10_SYMBOL) is equal to a default tag (The comparison function can be implemented by software or hardware).

Therefore, when the loaded predetermined tag matches the default tag, the result of the examining process is correct and the command can be transferred. On the other hand, the result of the examining process is incorrect, and the computer system 30 requires being debugged or re-booted.

[0040] In summary, a method embodiment including all characteristics of the present invention can refer to Fig.8, which is a flow chart showing the method for transferring a command among a plurality of devices of the computer system 30. Eachdevice corresponds to an application code.

[0041] Step300:start;

[0042] Step302:execute an examining process when a command is transferred so as to examine whether an application code of a device that is about to receive the command is correct or not. For instance, a predetermined instruction or a predetermined tag contained in theapplication code can be used to determine if the predetermined instruction or the predetermined tag conforms to a predetermined condition to determine the correction of the application code. If the predetermined instruction or the predetermined tag conforms to the predetermined condition, the

result of the examining process is correct with correct application code, and Step 304 is executed; otherwise, proceed with Step 306;

[0043] Step304:Permit transferring the command;

[0044] Step306:Do not transfer thecommand and proceed with a debug or a re-boot operation.

[0045] The present invention can be applied to a computer system whenever a command is transferred among a plurality of devices in the computer system. The characteristic of the present invention is to execute an examining process before the command is transferred so as to ensure that the command is transferred correctly and to avoid any abnormal operation of the computer system. When thecharacteristic of the present invention is applied to the boot/reboot process and the mode transformation of the computer system, the computer system can be quickly and correctly relieved from the standby mode.

[0046] Those skilled in the art will readily observe that numerous modifications and alterations of the device and method may be made while retaining the teachings of the invention. Accordingly, the above disclosure should be construed as limited only by the metes and bounds of the appended claims.